



Module Outline

ITS 2122

Python for Data Science & AI

Version 1.2

1.0 Change History

Any changes or revisions made to the syllabus are documented here, including the details of the latest version in this document.

Version	Date	Revisions / Changes made
Version 1.0	October 2024	Released
Version 1.1	May 2025	Released
Version 1.2	August 2025	Released

2.0 Course Description

This intensive Python programming course equips students with essential programming skills focused on data science and AI applications. Starting with Python fundamentals, students quickly advance to working with key data science libraries including NumPy and Pandas, while learning crucial concepts in data manipulation and analysis. Through hands-on projects, participants develop practical skills in data processing, visualization, and basic machine learning implementations. By course completion, students will have the strong Python foundation needed to pursue advanced topics in data science and AI.

3.0 Course Objectives

By the end of this course, students will be able to:

- 1. Understand Python Basics:** Develop a foundational understanding of Python syntax, variables, and data types.
- 2. Control Flow Mastery:** Learn to implement decision-making and loops to control program flow effectively.
- 3. Function Design:** Write reusable and modular code by defining and using functions and understanding scope.
- 4. Data Structures and Manipulation:** Work with Python's built-in data structures (lists, tuples, dictionaries, sets) to manage and manipulate data efficiently.

5. **File Handling:** Gain proficiency in reading from and writing to files for data persistence.
6. **Data Analysis with NumPy and Pandas:** Leverage NumPy for numerical computations and Pandas for data manipulation, cleaning, and analysis using DataFrames and Series.
7. **Visualization and Data Access:** Create insightful data visualizations using Matplotlib, and acquire data through web scraping techniques and API integration for real-world data processing tasks.

4.0 Prerequisite

ITS 1010 - Programming Fundamentals

5.0 Module Details & Credits

- **Module Code:** ITS 2122
- **Module Credits:** 3.0
- **Total Lecture Hours:** 64 Hours
- **Expected Workload:** This credit value is based on the 64 total hours of lectures plus an estimated 8-10 hours of independent work per week (including assignments, practical exercises, and self-study) over the duration of the module.

6.0 Syllabus Detail

Module 1 Introduction to Python and Basic Syntax	Introduction to Programming and Python <ul style="list-style-type: none">• What is programming?• Why Python?• Installing Python and IDE setup• Writing first program: "Hello, World!" Python Basics <ul style="list-style-type: none">• Variables and data types• Basic operators• Input/output operations• Type conversion• Comments and documentation Practical Exercises <ul style="list-style-type: none">• Basic calculations• User input handling• Simple text manipulation Control Flow <ul style="list-style-type: none">• if, elif, else statements• Match-case statement• Comparison operators• Logical operators• Truth values Loops <ul style="list-style-type: none">• while loops• for loops
---	---

	<ul style="list-style-type: none"> • break and continue • range() function <p>Practice Problems</p> <ul style="list-style-type: none"> • Decision-making programs • Number guessing game • Simple calculator
<p>Module 2</p> <p>Data Structures</p>	<p>Lists</p> <ul style="list-style-type: none"> • Creating and accessing lists • List methods • List slicing • List comprehensions <p>Tuples and Sets</p> <ul style="list-style-type: none"> • Tuple operations • Set operations • Converting between data structures • When to use each type <p>Dictionaries</p> <ul style="list-style-type: none"> • Creating dictionaries • Accessing and modifying • Dictionary methods • Nested dictionaries <p>Practice Session</p> <ul style="list-style-type: none"> • List manipulation exercises • Dictionary-based programs

	<ul style="list-style-type: none"> • Data structure conversion • Real-world applications
<p>Module 3</p> <p>Functions and Modules</p>	<p>Functions</p> <ul style="list-style-type: none"> • Function definition • Parameters and arguments • Return values • Default arguments • Variable scope <p>Important built-in functions</p> <ul style="list-style-type: none"> • Some math functions • map() & filter() functions <p>Lambda Functions</p> <ul style="list-style-type: none"> • Basic syntax and example problems • Practical applications <p>Modules and Packages</p> <ul style="list-style-type: none"> • Using built-in modules • Creating custom modules • import statements • pip and package management • Building custom packages

Module 4 Object-Oriented Programming	OOP Concepts <ul style="list-style-type: none">• Classes and objects• Attributes and methods• Constructors• self keyword Inheritance <ul style="list-style-type: none">• Single inheritance• super() function• Method overriding Advanced OOP <ul style="list-style-type: none">• Polymorphism• Encapsulation
Module 5 File Handling	File Operations <ul style="list-style-type: none">• Reading files• Writing files• File modes• Context managers (with statement) Working with Different File Formats <ul style="list-style-type: none">• Text file processing• JSON files processing• CSV file processing

Module 6 Data Manipulation	NumPy Basics <ul style="list-style-type: none">• Arrays and operations• Mathematical functions• Array manipulation Pandas Introduction <ul style="list-style-type: none">• Series and DataFrames• Data loading• Basic operations• Data analysis
Module 7 Data Visualization	Matplotlib <ul style="list-style-type: none">• Anatomy of a plot• Common plot types• Customization• Subplots Seaborn <ul style="list-style-type: none">• Categorical Plots• Distribution Plots• Relational Plots

Module 8

APIs, Data Collection & Debugging

Web Scraping

- 'requests' library
- BeautifulSoup library
- HTML basics
- Data extraction

API Integration

- REST APIs
- JSON parsing
- API authentication
- Making API calls

Debugging Fundamentals

- Understanding error messages
- Common Python errors and solutions
- Using print statements effectively
- Logging for debugging
- Python debugger (pdb)

7.0 Evaluation Process

In-Class Quizzes: 25% of total grade

- Short quizzes conducted during class sessions to reinforce and assess understanding of key concepts
- Designed to encourage consistent engagement and retention.

Midterm Exam: 30% of total grade

- A comprehensive examination covering all material up to the midpoint of the course.

Final Group Project and Presentation: 40% of total grade

- Students will work in small groups to design, implement, and evaluate an end-to-end data science solution using real-world datasets.
- The project will culminate in a presentation showcasing the problem tackled, methodology, results, and deployment considerations.
- Assessed on technical depth, creativity, teamwork, and communication.

Class Participation: 5% of total grade

- Participation in class discussions, attendance, and engagement during lectures and labs.

8.0 Module Reference Materials

1. Python Official Documentation (<https://docs.python.org/>)
2. "Python Crash Course, 2nd Edition" by Eric Matthes (2019) [Read online: [click here](#)]
3. "Fluent Python: Clear, Concise, and Effective Programming, 2nd Edition" by Luciano Ramalho (2019) [Read online: [click here](#)]
4. "Python for Data Analysis, 3rd Edition" by Wes McKinney (2019) [Read online: [click here](#)]
5. Real Python (<https://realpython.com/>)